# 3.1 Introducción a Wireshark

Una herramienta básica para observar los mensajes intercambiados entre aplicaciones es un **analizador de protocolos** (packet sniffer). Un analizador de protocolos es un elemento pasivo, únicamente observa mensajes que son transmitidos y recibidos desde y hacia un elemento de la red, pero **nunca** envía él mismo mensajes. En su lugar, un analizador de protocolos recibe una copia de los mensajes que están siendo recibidos o enviados en el terminal donde está ejecutándose.

Está compuesto principalmente de dos elementos: una **librería de captura de paquetes,** que recibe una copia de cada trama de enlace de datos que se envía o recibe, y un **analizador de paquetes**, que muestra los campos correspondientes a cada uno de los paquetes capturados. Para realizar esto, el analizador de paquetes ha de conocer los protocolos que está analizando de manera que la información mostrada sea coherente. Es decir, si capturamos un mensaje HTTP, el analizador de paquetes ha de saber que este mensaje se encuentra encapsulado en un segmento TCP, que a su vez se encuentra encapsulado en un paquete IP y éste a su vez en una trama de Ethernet. Un esquema de la integración de Wireshark con el Sistema Operativo del equipo en uso puede verse en la Imagen 1.

Wireshark se trata de un software gratuito disponible para varias plataformas (Unix, Windows y Mac OS). Se puede descargar desde www.wireshark.org., donde además, existe documentación asociada y un manual de usuario.

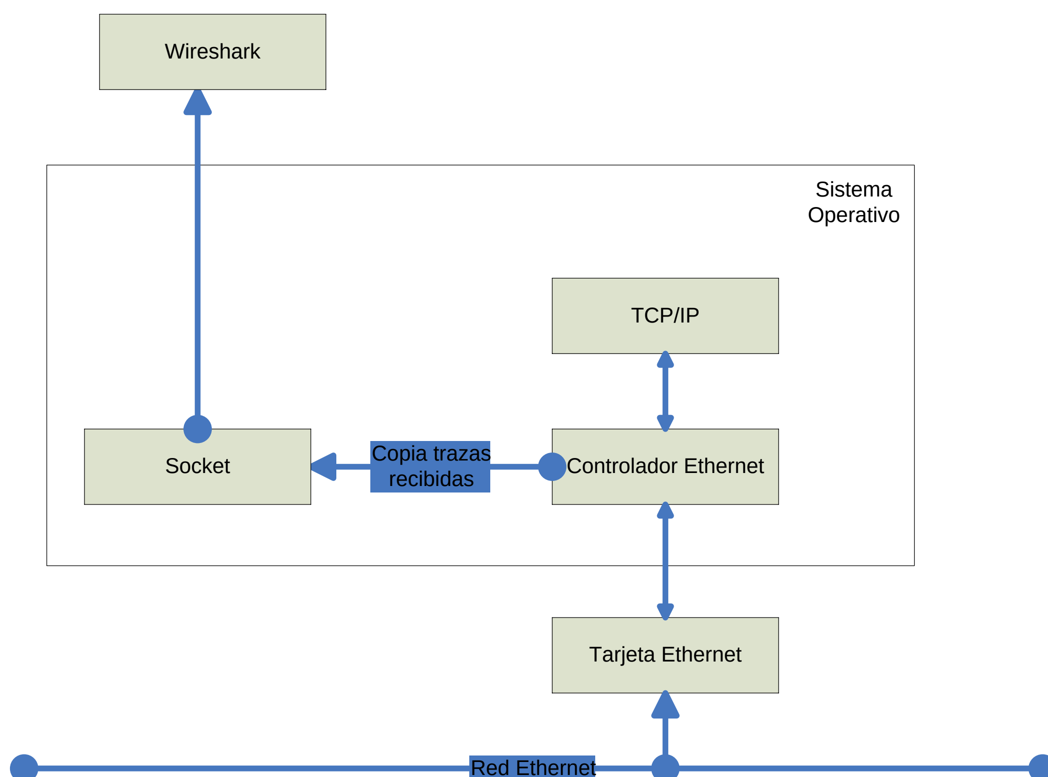



**Imagen 1 - Esquema integración Wireshark con el SO.**

# 3.1.1 Getting Started

En los equipos de laboratorio se encuentra instalado el analizador de paquetes (Wireshark) y las librerías de captura de paquetes (WinPcap). Al ejecutar el acceso directo de Wireshark en el escritorio debería e aparecer una pantalla como la siguiente:

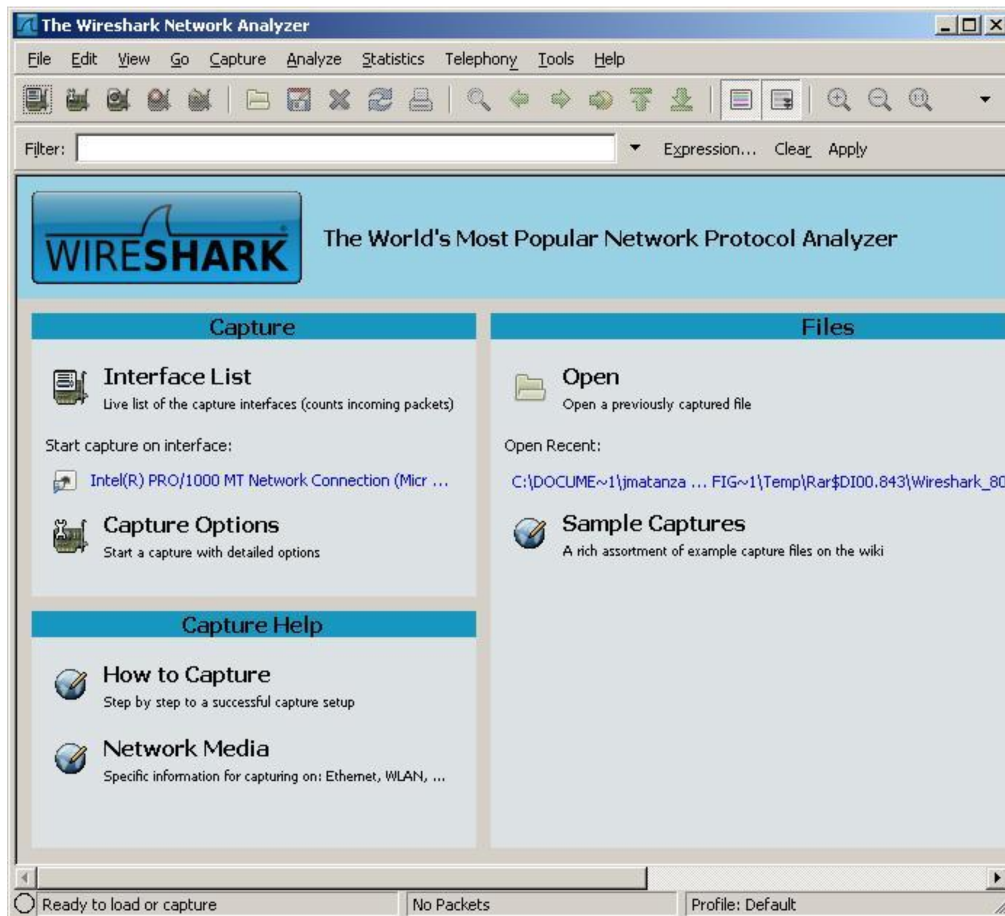



**Imagen 2 - Pantalla inicial Wireshark**

Con los iconos de arriba a la derecha podemos controlar las capturas. Si pulsamos sobre el de más a la izquierda, veremos las posibles interfaces de red desde donde podemos capturar (Únicamente podremos capturar desde una interfaz de red al mismo tiempo). La pantalla es algo así:

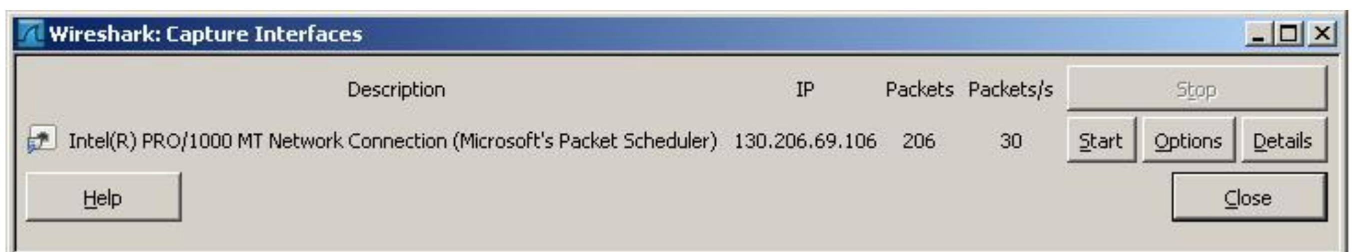



**Imagen 3 - Interfaces de red disponibles para la captura**

En el caso de que existan más interfaces de red, éstas serán listadas. Si pulsamos el botón "Start" comenzaremos con la captura. Veremos cómo Wireshark comienza a mostrar todos los mensajes de red enviados/recididos.

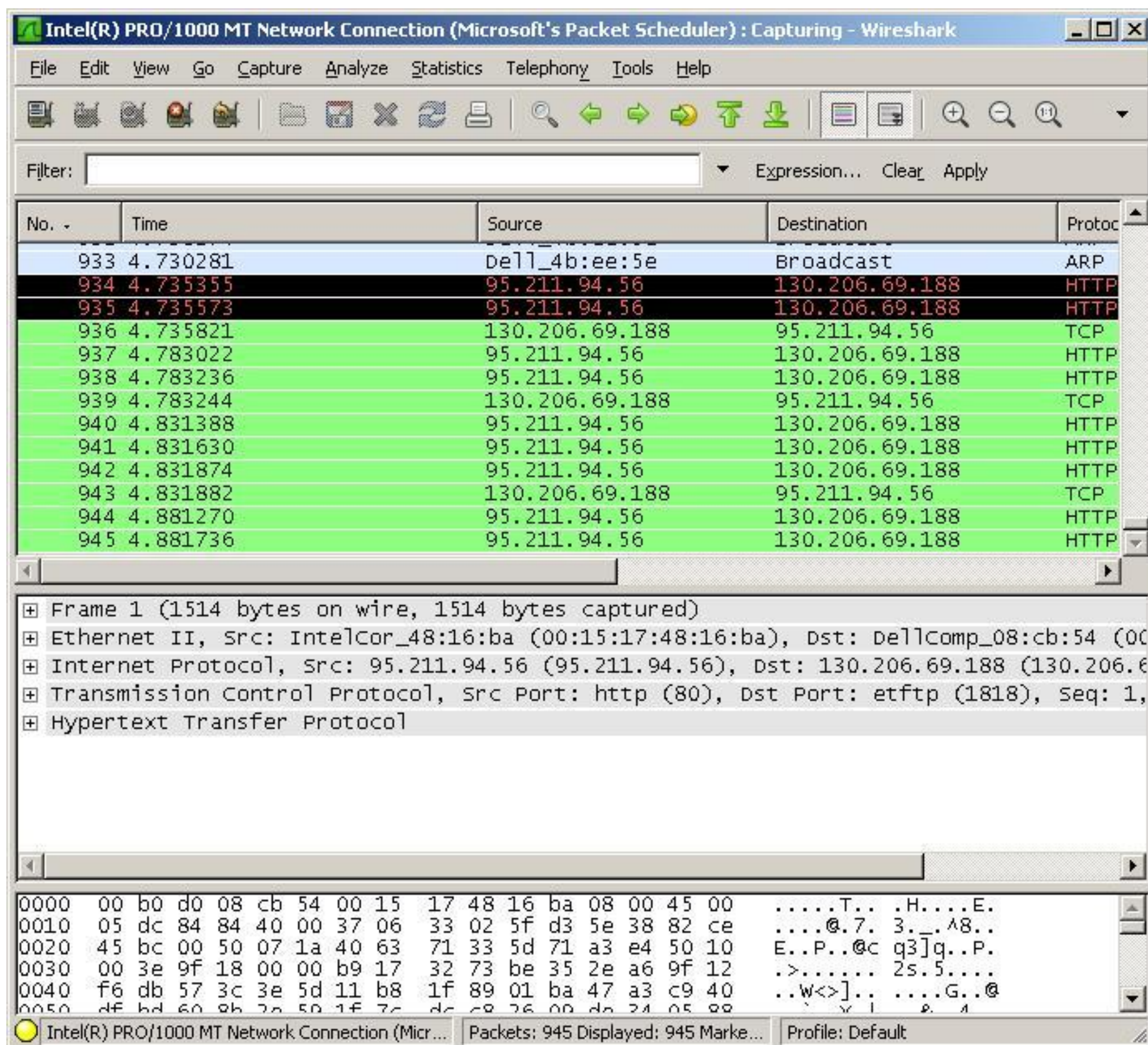



**Imagen 4 - Captura de paquetes.**

Una de las principales utilidades de Wireshark es la posibilidad de filtrar los mensajes en tiempo real. Veámoslo con un ejemplo.

1. Abrir un navegador de internet.
2. Con Wireshark capturando paquetes, filtrar con "dns" (sin comillas).
3. Acceder a una página web.

Intentad analizar los mensajes intercambiados:

1. *¿Qué tipos de mensajes se intercambian?*
2. *Interpretad la respuesta.*

# 3.1.2 Análisis de una consulta HTTP

A continuación como veremos, podemos utilizar Wireshark para estudiar el intercambio de mensajes que se realiza a la hora de realizar una consulta http a una página web sencilla.

1. En primer lugar abrimos un navegador web que esté instalado en la máquina.
2. Abrimos Wireshark y seleccionamos la interfaz de red adecuada. Como ya hemos observado anteriormente, se muestra todo el tráfico de red cursado por el PC.
3. Le indicamos que únicamente nos filtre los mensajes correspondientes al protocolo http. Escribimos "http" sin comillas en el campo correspondiente al texto y pulsamos el botón "Apply" para que comience a filtrar. En este momento únicamente filtrará mensajes pertenecientes al protocolo http.
4. En el navegador, accedemos a la siguiente dirección de red: [http://192.168.56.200/part1.html](http://192.168.56.200/part1.html) . El navegador deberá mostrar una página web con algún texto.
5. Paramos la captura de Wireshark.

En la pantalla de Wireshark debemos de tener algo parecido a lo siguiente:

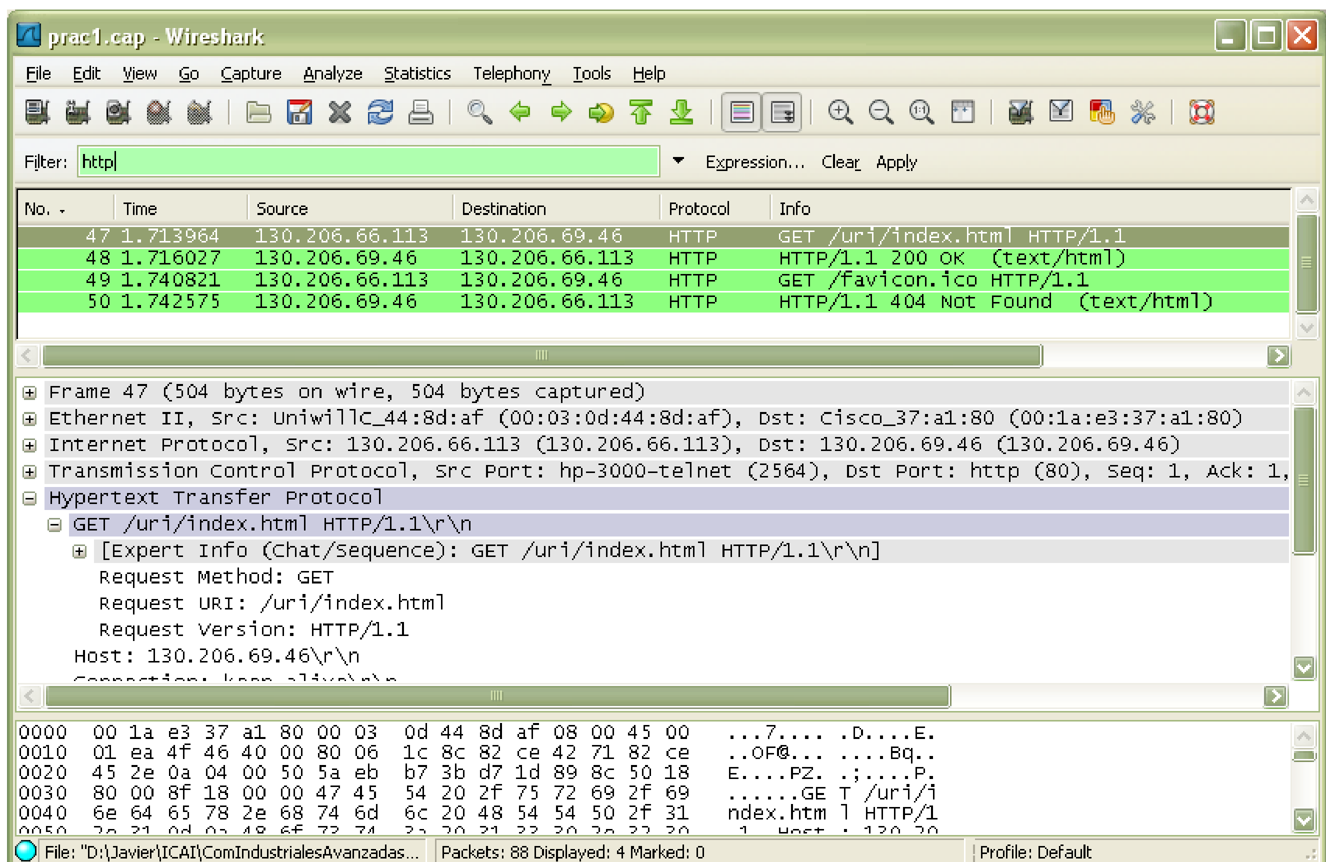



**Imagen 5 – Pantalla Wireshark tras haber capturado una consulta http sencilla.**

En el ejemplo de la Imagen 5 podemos ver ágilmente el encapsulamiento que se ha llevado a cabo para enrutar el mensaje http. Wireshark muestra el mensaje http que se

encapsuló en un segmento TCP, que se encapsuló en un paquete IP y que se encapsuló en una trama de ethernet. Ya que únicamente estamos interesados en el mensaje http, comprimiremos el resto de información.

Observamos cómo se ha enviado una petición ("GET") a la url solicitada al navegador. Tras ésta, el servidor web donde se encuentra alojada la página ha contestado satisfactoriamente (200 OK) encapsulando en un mensaje http el código html que contiene la ruta requerida. Es el navegador (aplicación) quien desencapsula el código y lo interpreta.

Se observará también la existencia de una petición contigua en la uri: favicon.ico. Esto se debe a que el navegador hace una consulta acerca de si el servidor tiene algún icono que pueda mostrar junto a la ruta. Ya que nuestro servidor no tiene dicho icono, se contesta con el código correspondiente (404 Not Found).

La cabecera del mensaje http contiene alguna información que puede ayudar al navegador a interpretar el mensaje encapsulado. Intentad averiguar lo siguiente:

3. *¿Qué idioma tiene definido el navegador? ¿Para qué puede querer el servidor web conocer esta información?*
4. *¿Cuándo fue la última vez que se realizó algún cambio en la página solicitada? ¿Para quién es útil saber esta información (Navegador o servidor)? ¿Por qué?*
5. *¿Cuál es el tamaño del "payload" encapsulado en http?*
6. *Sin mirar en una tabla ASCII, ¿qué codificación hexadecimal tiene el carácter "G"?(Pista: hay una "G" en "GET HTTP").*
7. *¿Qué valor tiene el CRC(Hexadecimal)?*

# 3.1.3 Consulta HTTP condicional

En este apartado veremos cómo se realiza el diálogo de una consulta http condicional.

En primer lugar es necesario limpiar la caché del navegador que estemos utilizando (Para Netscape 7.0, Edicion->Preferencias->Avanzado->Cache, limpiar caché. Para Firefox, Herramientas->Limpiar datos privados. Para IE, Herramientas->Opciones de Internet->Eliminar archivo. Para Chrome, Eliminar Datos de Navegación.)

A continuación realizamos los siguientes pasos:

1. Abrimos un navegador web con la caché limpia.
2. Abrimos Wireshark y filtramos el tráfico http.
3. Usando el navegador, accedemos a la dirección anteriormente usada (http://192.168.56.200/part1.html).
4. Una vez se haya cargado la página, pulsamos el botón "Refrescar" en el navegador (F5).
5. Paramos la captura de Wireshark.

Habremos obtenido algo similar a lo siguiente:

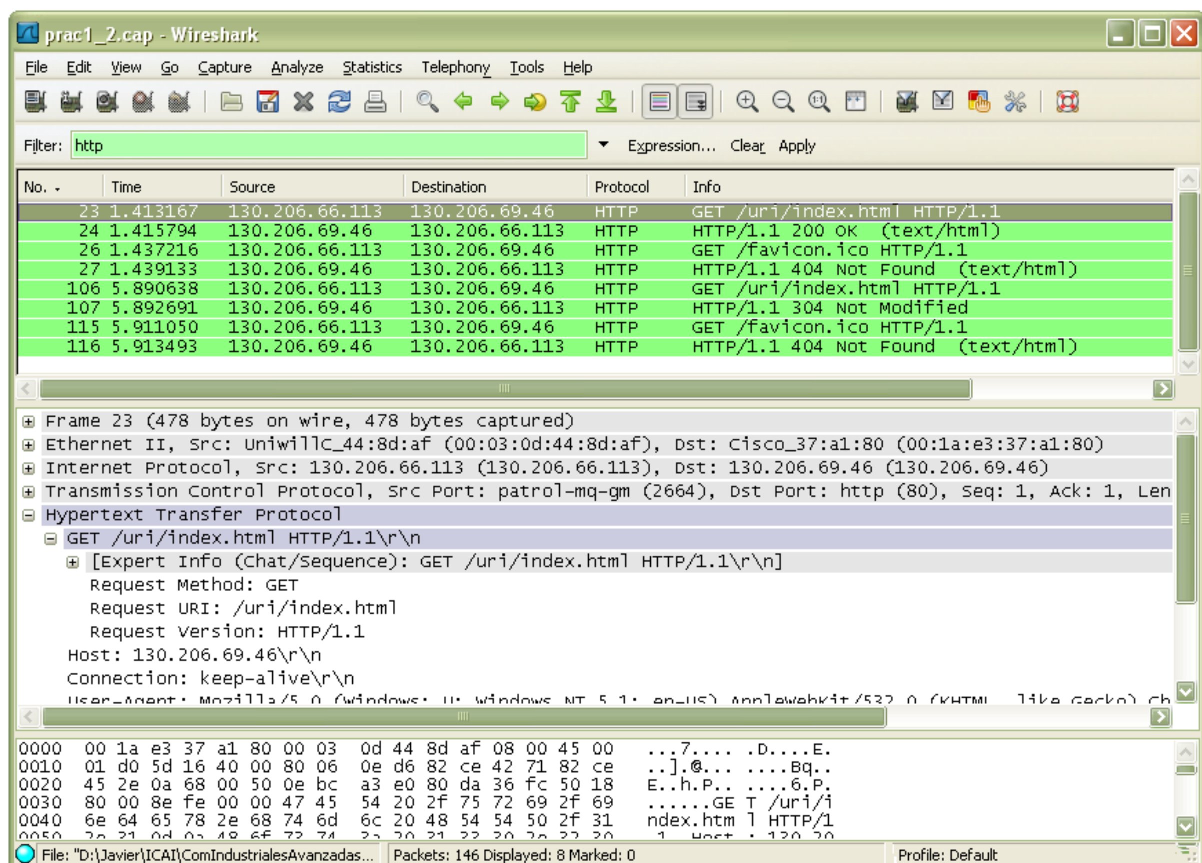



**Imagen 6 - Pantalla Wireshark tras haber capturado una consulta http condicional**

Ignorando de nuevo las peticiones favicon.ico, a primera vista parece que los mensajes GET son idénticos, pero observando más detenidamente el encabezado de http, veremos que no es así. Además, en ambos casos la respuesta del servidor web es completamente diferente.

Intentad averiguar lo siguiente:

8. *¿Se encuentra la cabecera de “If-Modified-Since” en ambos mensajes GET http? ¿A qué crees que de debe?*
9. *¿Cuál es el Status-Code devuelto por el servidor tras la segunda consulta?*
10. *¿Se devolvió el contenido de la página en tras la primera consulta?¿Y tras la segunda?¿Cómo puedes verlo?*

# 3.1.4 Consultas de archivos con gran longitud

Hasta ahora hemos estudiado el intercambio de mensajes cortos en un entorno Cliente-Servidor. Veremos ahora una potente ayuda que Wireshark proporciona a la hora de analizar relaciones entre paquetes.

1. Abrir un navegador web.
2. Ejecutar Wireshark sin aplicar ningún tipo de filtrado.
3. Acceder a la siguiente dirección utilizando el navegador: http://192.168.56.200/part2.html).
4. Parar la captura de Wireshark.

Si filtramos ahora por http, veremos como no existe diferencia alguna con la consulta/respuesta realizada en el apartado 1. No obstante, mientras en el apartado 1, el archivo que nos devolvía el servidor tenía un tamaño aproximado de 165 bytes, en este caso el tamaño del archivo es de 30KBytes. Este detalle es irrelevante en la capa correspondiente a http, pero en capas inferiores es necesario gestionarlo de alguna manera.

Filtremos ahora la captura con "tcp". Veremos algo parecido a lo siguiente:

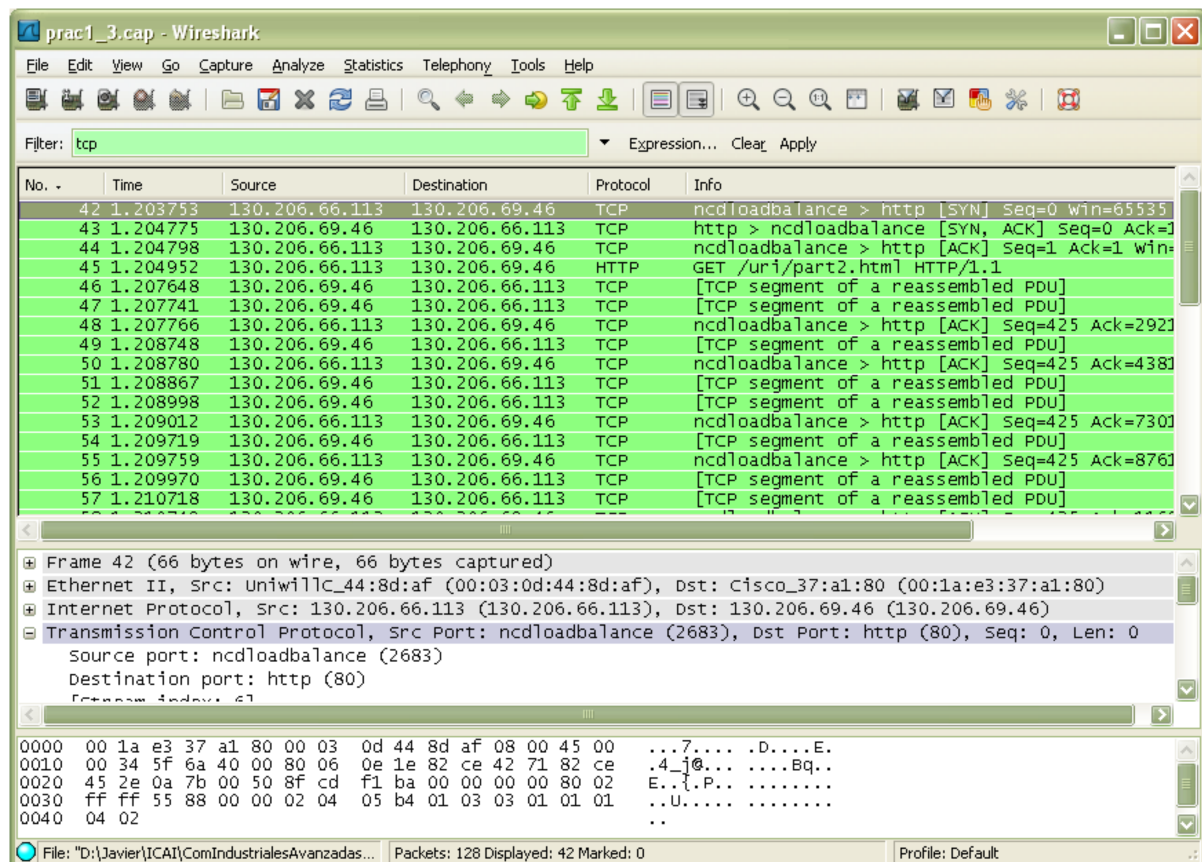



**Imagen 7 - - Pantalla Wireshark tras haber capturado una consulta http "larga"**

TCP entre el cliente y el servidor.

Efectivamente, la respuesta http del servidor es descompuesto en varios segmentos TCP que se envían de manera secuencial. Wireshark nos muestra todos los paquetes y nos indica el lugar en el que se reconstruirá el mensaje que encapsulan:

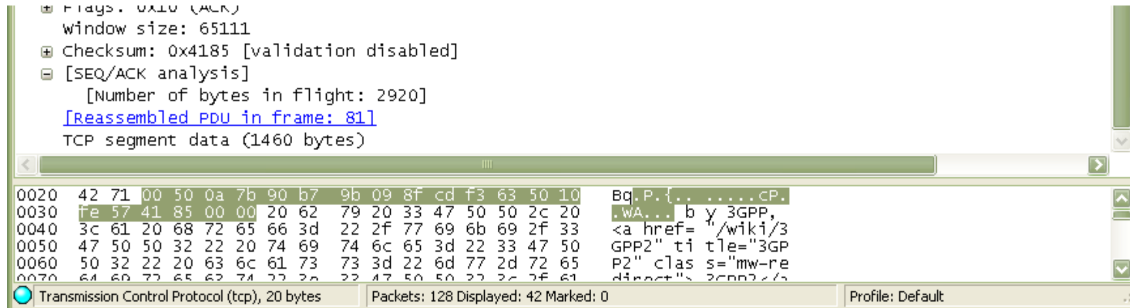



**Imagen 8 - Detalle información del paquete TCP indicando cuando se reconstruirá el mensaje fragmentado en la PDU.**

11. *¿En cuántos segmentos TCP se ha fragmentado la información?*
12. *¿Por qué hay segmentos TCP en sentido Cliente->Servidor?*
13. *¿Qué función correspondiente un protocolo de Enlace de Datos realizan estos paquetes?. Explica las funciones de los campos "Acknowledgement number" y "Sequence number".*